LAB=02
PRIMALITY TESTING

```
#include <iostream>
using namespace std;
float count=0;
int gcd(int a,int b)
{
   if(a<b)
   {
      return gcd(b,a);
   }
   else if(a%b==0)
   {
      return b;
   }
   else
   {
      return gcd(b,a%b);
   }
}
int power(int a,unsigned int x,int p)
{
   int res=1;
   a = a%p;
   while(x>0)
   {
      if(x & 1)
      {
         res = (res*a)%p;
      }
      x=x/2;
      a=(a*a)%p;
   }

   return res;
}
bool isPrime(unsigned long int n,int k)
{
   if(n<=1 || n==4)
   {
      return false;
```

```
    }
    if(n<=3)
    {
        return true;
    }
    while(k>0)
    {
        int a = 2 + rand() % (n-4);
        if(gcd(n,a)!=1)
        {
            return false;
        }
        if(power(a,n-1,n)!=1)
        {
            return false;
        }
        for(a=2;a<=n;a++)
        {
            int
            r=power(a,n-
            1,n);if(r==1)
            {
                count++;
            }
        }
        k--;
    }


     return true;
}
int main()
{
    int k=3;
    if(isPrime(1009,k)==t
    rue)
    {
        cout<<"Prime";
    }
    else
    {
        cout<<"Composite";
    }
     cout<<endl<<"Count:
    "<<count;return 0;
}
```

Output: